Photonic Inverse Design Using the Adjoint Method

- **Lumopt**\(^1\) Python module for adjoint sensitivity analysis
- FDTD Solutions for 2D/3D simulation
- SciPy gradient based optimization algorithms

= Highly efficient optimization of photonic components

Try yourself: Examples and software [lumeri.ca/ofc](http://lumeri.ca/ofc)

\(^1\) [github.com/chriskeraly/lumopt](https://github.com/chriskeraly/lumopt)

See more @Booth 5438
Motivation

- Component design challenging, even for basic components
- We would like a lot:
  - No reflections
  - No loss
  - Insensitive to manufacturing imperfections
  - Works for range of wavelengths
  - Works at different temperatures
- Usually no analytic solution
- Good solutions using PSO
- Can we do better with adjoint methods?

https://github.com/lukasc-ubc/SiEPIC_EBeam_PDK
Lumerical’s Suite of Simulation Tools for Photonics

This demo uses FDTD simulation automated via Python API
Lumopt: Python Based Inverse Design for Lumerical FDTD

- Lumopt: open source adjoint sensitivity analysis
- Collaboration with Lumerical over past year
- Targets integrated photonics
- Now included with FDTD Solutions

https://github.com/chriskeraly/lumopt

Optics Express, Vol 21, Issue 18, 2013
Inverse Design vs Forward Design

Traditional Design

Inverse Design

3dB Power Splitter

Costly Iterations

FOM
IL = 0.3dB

FOM
IL = 0.1dB

Adjoint optimization tool
Parametric Shape based adjoint optimization

**Parametric shape**
- Defines design space
- Optimization parameters

**Adjoint sensitivity analysis**
- Efficiently compute gradient
- 2 FDTD simulations
- Independent of # parameters

**Gradient based optimization**
- Highly efficient optimization
- Uses more physics of device

https://hackernoon.com/gradient-descent-aynk-7cbe95a778da
Example: Full component design flow for Y-Branch

Lumopt in Action
Full component design flow for Y-Branch

- Objective: build a splitter like prior art below
- Use inverse design to build splitter section (1)
- Add waveguide offset arms (2) post-optimization

A compact and low loss Y-junction for submicron silicon waveguide

https://github.com/lukasc-ubc/SiEPIC_EBeam_PDK
An Inverse Design Flow

Step 1
Define waveguide position & optimization monitors
Base simulation script file

Step 2
Define polygon location and parameters to be optimized
Parametrized polygon

Step 3
Run adjoint optimization in 2D FDTD mode
Interim best parameters

Step 4
Run adjoint optimization in 3D FDTD
Best parameters (final)

Step 5
Extract optimized parameters
Export component shape into GDSII

Lumerical script → Polygon geometry → Initial parameters → Best parameters
Step 1: Define base simulation

2D simulation
- Uses an effective index for waveguide
- Good approximation
- Fast simulation

Base simulation is defined by Lumerical Script (lsf)
Step 2: Define parametric shape

- Parametric shape defined as Python function
- Function argument is list of parameters
- Function returns list of polygon vertices

```python
def taper_splitter(params = np.linspace(0.25e-6, 2e-6, 20)):
    ... Defines a taper where the parameters are the y coordinates of the nodes of a cubic
    points_x = np.concatenate([[-2.51e-6], np.linspace(-2.5e-6, 2.5e-6, 20), [2.51e-6]])
    points_y = np.concatenate(([0.25e-6], params, [2e-6]))
    n_interpolation_points = 100
    px = np.linspace(min(points_x), max(points_x), n_interpolation_points)
    interpolator = sp.interpolate.interp1d(points_x, points_y, kind = 'cubic')
    py = interpolator(px)
    py = np.minimum(2.5e-6, py)
    py = np.maximum(np.concatenate((np.ones(50)*0.2e-6, np.ones(50)*0.53e-6)), py)
    px = np.concatenate((px, px[40:][::-1]))
    py = np.concatenate((py, py[40:][::-1]-0.5e-6))
    polygon_points_up = [(x, y) for x, y in zip(px, py)]
    polygon_points_down = [(x, -y) for x, y in zip(px, py)]
    polygon_points = np.array(polygon_points_up[::-1] + polygon_points_down)
    return polygon_points
```
Step 3: Run fast 2D optimization

This optimization runs in 20-30 minutes
Step 4: Refine with 3D optimization

- This step is largely the same as 2D simulation
- Takes a bit longer to run
- Should complete with few iterations if seeded with 2D solution
Step 5: Save design to GDSII

- Optimized shape and output arms saved to GDSII
- Similar to prior art, but has a few ripples!
Step 5: Y splitter example: Compact model extraction from layout

- Import the final GDSII mask into 3D simulation
- Define ports
- Extract the S-parameters
- Save to data file for INTERCONNECT circuit simulation
Example: Broadband & Compact Y-Branch
Broadband & Compact Splitter

- Can we make a smaller splitter?
- Can we ensure broadband?

- Parametric shape with output waveguides, 20 parameters
- 5x5 footprint footprint
- FOM taken over C+L bands

A compact and low loss Y-junction for submicron silicon waveguide

https://github.com/lukasc-ubc/SiEPIC_EBeam_PDK
Broadband Inverse Design

- Optimize FOM over a spectrum
- No additional FDTD simulations required!
Step 3: Run fast 2D optimization

This example takes ~60 minutes to run:

FOM = 0.5 = ideal
Example: Grating coupler

Available soon!!
Double etch grating coupler

Example available soon

>80 optimization parameters
Example: Robust Y-Branch

Lumopt in Action
Co-optimization:

- Run multiple optimizations concurrently
- Optimizations share same parameters
- Figure of merit or structure can be different

Example uses:

- Dual polarization devices (different FOM)
- Multiple wavelengths (different FOM)
- Optimize process corners (different geometry)

\[ \text{FOM}(p) = \sum_{i=1}^{N} FOM_i(p) \]

\[ p = (p_1, \ldots, p_N) \]
Co-optimization: Robust splitter

- Build a splitter tolerant to manufacturing error
- Co-optimize 2 different shapes (same parameters)
- “Over etch” slightly smaller than nominal
- “Under etch” slightly larger than nominal
- Same FOM function
Co-optimization: Robust splitter

- Setup 2 optimizations
- Sum the figures of merit
- 2 FDTD simulations/FOM/iteration

Co-optimization of +/- 20nm on edge position

Nice smooth shape!
Upcoming features
Layout using Cadence’s Virtuoso CurvyCore Technology

- Non-Manhattan shapes
- Symbolic equations provide accurate mathematical model
- Generates high-quality polygon representations for fabrication
- Ideal for inverse design shapes
Two Approaches to Inverse Design

Parametric Geometry Optimization
- Finds optimal parameters for shape(s)
- Parametric shape defines/limits design space

Topology Optimization
- User provides footprint and 2 materials
- No intuition about shape required!
- Solver finds best solution

Available Now!

Coming Soon!

$$\epsilon_{low} = 1.44$$
$$\epsilon_{high} = 2.88$$
Topology Optimization

Supports:
• Broadband
• Quasi-2D
• Constrained feature size
• Co-optimization
Topology Optimization: Broadband (1450-1650nm) TE Splitter
Next Steps

Try running examples
• Stick to 2D, get results in minutes
• Set max_iter=3 to get suboptimal device fast

Try some modifications
• Change device footprint
• Change bandwidth
• Change number of optimization parameters
• Try pCell suggestions in tutorial

More examples available in applications gallery

See more
Exhibit Hall Booth 5438